

SQL 활용 통합 실습

1. 다음은 웹 사이트의 게시판을 사용하는 회원을 관리하기 위한 회원 테이블 인스턴스입니다.. 회원 테이블 인스턴스를 참고하여 테이블을 생성하시오.

테이블명 : MEMBER

열이름	데이터타입(길이)	제약조건	제약조건이름	설명
USER_ID	NUMBER(6)	Primary key	-	회원번호
USER_NAME	VARCHAR2(20)	Not Null	member_uname_nn	회원이름
PASSWD	VARCHAR2(10)	Not Null	member_pw_nn	암호
ID_NUM	VARCHAR2(13)	Unique Not Null	member_idnum_uk member_idnum_nn	주민등록번호
PHONE	VARCHAR2(13)			전화번호
ADDRESS	VARCHAR2(25)			주소
REG_DATE	DATE			가입일
INTEREST	VARCHAR2(15)			관심분야

CREATE TABLE member

(USER_ID NUMBER(6) primary key ,

USER_NAME varchar2(20) constraint member_uname_nn not null,

PASSWD varchar2(10) constraint member_pw_nn not null,

ID_NUM varchar2(13) constraint member_idnum_uk unique

Constraint member_idnum_nn not null,

PHONE varchar2(13),

ADDRESS varchar2(25),

REG_DATE date,

INTEREST varchar2(15));

2. 다음은 웹 사이트의 게시판을 사용하는 회원을 관리하기 위한 게시판 테이블 인스턴스입니다. 인스턴스를 참고하여 테이블을 생성하시오.

테이블명 : BOARD

열이름	데이터타입(길이)	제약조건	제약조건이름	설명
NO	NUMBER(4)	Primary key	-	게시글번호
SUBJECT	VARCHAR2(50)	Not Null	Board_sub_nn	제목
CONTENT	VARCHAR2(2000)			내용
CRE_DATE	DATE			작성일
USER_ID	NUMBER(6)	Foreign Key	Board_uid_fk	MEMBER(USERID)참조

CREATE TABLE board

(NO number(4) primary key,

SUBJECT varchar2(50) constraint board_sub_nn not null,

CONTENT varchar2(2000),

```
CRE_DATE date,
```

```
USER_ID number(6) constraint board_uid_fk references MEMBER(USER_ID));
```

3. 게시판의 NO 열에 사용할 board_no_seq라는 이름의 시퀀스를 생성하시오. 시퀀스의 시작번호와 증분값은 각각 1로 설정하고 NOCACHE, NOCYCLE 속성을 주시오.

```
CREATE SEQUENCE board_no_seq
```

```
INCREMENT BY 1
```

```
START WITH 1
```

```
NOCACHE
```

```
NOCYCLE;
```

4. 회원 테이블과 게시판 테이블을 참조하여 무결성 제약조건을 위반하지 않는 데이터를 3건씩 입력하시오. 게시판의 번호는 board_no_seq 시퀀스를 사용하여 입력하시오.

```
INSERT INTO member
```

```
VALUES ('101','송성광','비밀번호','7906021234567', '051-123-1234','부산 수정동',sysdate,'DB');
```

```
INSERT INTO member
```

```
VALUES ('102','김영균','비밀번호1','7903022341567','051-321-1234','창원 사림동',sysdate,'internet');
```

```
INSERT INTO member
```

```
VALUES ('103','전인하','비밀번호2','7901041324668','051-345-3456','부산 동삼동',sysdate,'java');
```

```
INSERT INTO board
```

```
VALUES (board_no_seq.nextval,'제목1','내용1',sysdate,'101');
```

```
INSERT INTO board
```

```
VALUES (board_no_seq.nextval,'제목2','내용2',sysdate,'102');
```

```
INSERT INTO board
```

```
VALUES (board_no_seq.nextval,'제목3','내용3',sysdate,'103');
```

5. 회원 테이블에 email 열을 추가하시오. 단, email 열의 데이터 타입은 VARCHAR2(50)이다.

```
ALTER TABLE member ADD email VARCHAR2(50);
```

6. 회원 테이블에 국적을 나타내는 country 열을 추가하시오. 데이터타입은 VARCHAR2(20), 기본값은 'Korea'로 지정하시오.

```
ALTER TABLE member ADD country VARCHAR2(20) DEFAULT 'Korea';
```

```
SELECT * FROM member;
```

7. 회원 테이블에서 ID_NUM 열을 삭제하시오.

```
ALTER TABLE member DROP COLUMN id_num;
```

8. 회원 테이블의 address 열의 데이터 크기를 50으로 변경하시오.

```
DESC member;
```

```
ALTER TABLE member MODIFY address VARCHAR2(50);
```

9. 게시판 테이블의 user_id 열에 대해 board_userid_ix 라는 이름의 인덱스를 생성하시오.

```
CREATE INDEX board_userid_ix ON board(user_id);
```

10. 회원테이블에서 유저아이디, 이름, 전화번호, 주소를 볼 수 있는 member_addr_phone_list_vu라는 이름의 뷰를 생성하시오.

```
CREATE OR REPLACE VIEW member_addr_phone_list_vu
```

```
AS
```

```
SELECT user_id, user_name, phone, address
```

```
FROM member;
```

11. 게시판글번호, 제목, 유저아이디를 볼 수 있는 board_list_vu라는 이름의 뷰를 생성하시오.

```
CREATE OR REPLACE VIEW board_list_vu
```

```
AS
```

```
SELECT no, subject, user_id
```

```
FROM board;
```

12. member_addr_phone_list_vu와 board_list_vu라는 이름의 뷰에 대하여 각각 m과 b라는 이름의 동의어를 생성하시오.

```
CREATE SYNONYM m FOR member_addr_phone_list_vu;
```

```
CREATE SYNONYM b FOR board_list_vu;
```

13. board_list_vu 라는 뷰에서 작성일(CRE_DATE)가 보이도록 뷰를 수정하시오.

```
CREATE OR REPLACE VIEW board_list_vu
```

```
AS
```

```
SELECT no, subject, user_id, cre_date
```

```
FROM board;
```

14. USER_CONSTRAINTS, USER_INDEXES 및 USER_OBJECTS를 조회하여 제약조건과 사용자 소유의 객체의 이름, 타입, 상태 등을 조회하시오.

```
SELECT constraint_name, constraint_type  
  
FROM user_constraints  
  
WHERE table_name = 'MEMBER';
```

```
SELECT table_name, index_name  
  
FROM user_indexes  
  
WHERE table_name = 'MEMBER';
```

```
SELECT object_name, object_type, status  
  
FROM user_objects  
  
WHERE object_name LIKE 'MEMBER%' OR object_name='M'  
  
ORDER BY 2;
```

15. member테이블을 삭제하시오.

```
DROP TABLE member CASCADE CONSTRAINTS;
```

16. USER_OBJECTS를 조회하여 사용자 소유의 객체의 이름, 타입, 상태 등을 조회하면서 테이블 삭제의 결과를 정리하시오.

```
SELECT constraint_name, constraint_type  
  
FROM user_constraints  
  
WHERE table_name = 'MEMBER';
```

```
SELECT table_name, index_name  
  
FROM user_indexes  
  
WHERE table_name = 'MEMBER';
```

```
SELECT object_name, object_type, status  
  
FROM user_objects  
  
WHERE object_name LIKE 'MEMBER%' OR object_name='M'  
  
ORDER BY 2;
```

17. 모든 실습을 CLEAR 합니다.

DROP TABLE board;

DROP SEQUENCE board_no_seq;

DROP VIEW member_addr_phone_list_vu;

DROP VIEW board_list_vu;

DROP SYNONYM m;

DROP SYNONYM b;